A hybrid multi-scale approach for simulation of pedestrian dynamics

A. Kneidl[a], D. Hartmann[b], A. Borrmann[c]

[a] *kneidl@tum.de, Technische Universität München, 80290 München, Germany*
[b] *hartmann.dirk@siemens.com, Siemens Corporate Technology, 80200 München, Germany*
[c] *andre.borrmann@tum.de, Technische Universität München, 80290 München, Germany*

Abstract

One of the most important aspects for a realistic prediction of pedestrian flows is the modelling of human navigation in normal situations such as early design phases of buildings or transportation systems and hubs as well as in evacuation studies to enhance safety in existing infrastructures. To overcome the limitations of current navigation models, this paper proposes a new hybrid multi-scale model, which closely links information between the small-scale and large-scale navigation layer to improve the navigational behaviour. In the presented hybrid navigation model, graph-based methods using visibility graphs are used to model large-scale wayfinding decisions. The pedestrians' movements between two nodes of the navigation graph are modelled by means of a dynamic navigation field. The navigation field is updated dynamically during the runtime of the simulation, explicitly considering other pedestrians for determining the fastest path.

The proposed hybrid approach provides a realistic modelling of human navigational behaviour and thus a realistic prediction of flows since it reflects the human cognitive processes triggered by wayfinding tasks. This includes taking into account other pedestrians for routing decisions who are visible from the current position of the considered pedestrian. The paper discusses the concept and the technical details of the proposed hybrid multi-scale approach in detail and presents an extensive case study demonstrating its advantages.

Keywords: Pedestrian Dynamics, Navigation, Floor Field, Hybrid Model, Navigation Graph, Route Choice, Microscopic Model

1. Introduction

Studying emerging phenomena in pedestrian crowds is a classical subject in complexity science. Due to an improvement of algorithms as well as computational power, the simulation of pedestrian dynamics has attracted many researchers in the fields of safety and transportation science. On the one hand, pedestrian flow simulations are used for virtual evacuation studies (e.g. Lämmel et al., 2010; Klüpfel and Meyer-König, 2004; Rodriguez and Amato, 2010; Teknomo and Fernandez, 2012; Guo et al., 2011) and the proposal of corresponding optimal strategies (Hamacher and Tjandra, 2002) to enhance the safety of events and buildings.

On the other hand, simulation of pedestrian flows is an emerging topic in the early design phase of public transport systems, public transport hubs (Pelechano and Malkawi, 2008; Zhang et al., 2008) such as stations (Ding et al., 2011) or even large infrastructures and urban areas

(e.g. Loscos et al., 2003; Haklay et al., 2001) as well as their safety (Shi et al., 2012) and operation (Rindsfüser and Klügl, 2007). Of course, the utility of such virtual approaches depends crucially on the realism as well as on the computational speed of virtual approaches (Pelechano and Malkawi, 2008). If the realism of simulations is not sufficiently accurate, e.g. appearance of unnatural congestions, prediction of the simulations would yield improper designs. At the same time, if computations take too long, virtual approaches can hardly be used for design studies and design optimization.

From a conceptual point of view, models of pedestrian flows can be divided into microscopic models, simulating the behaviour of single pedestrians and their interaction, and macroscopic models, considering flows of pedestrian entities. Thus, being interested in local small-scale behaviour of pedestrian flows, microscopic models are the approach of choice. They can be distinguished into force models (e.g. Burstedde et al., 2001; Klüpfel, 2003; Kretz and Schreckenberg, 2006; Chraibi et al., 2011b), discrete-choice models (e.g. Antonini et al., 2006; Robin et al., 2009; Hoogendoorn and Bovy, 2004; Guo and Huang, 2012) or agent-based models (e.g. Rindsfüser and Klügl, 2007; Dijkstra et al., 2006; Ronald et al., 2007).

To study large-scale behaviour and large scale navigation, macroscopic models are typically preferred like network-based models (e.g. Hamacher and Tjandra, 2002; Mitchell and MacGregor Smith, 2001) or fluid dynamics models (e.g. Henderson, 1974; Helbing, 1992). Very often one is interested in local phenomena, e.g. the evolution of congestions, which are induced by large scale navigation, e.g. many people are choosing the same route.

This challenge, bridging the gap between small-scale and large-scale aspects can be resolved using multi-scale models, e.g. combining graph-based approaches either with force models (e.g. Wagoum et al., 2012; Kretz et al., 2011) or with agent type models (e.g. Funge et al., 1999; Reynolds, 1999; Rindsfüser and Klügl, 2007; Lerner et al., 2007; Sud et al., 2008; Teknomo, 2008; Asano et al., 2010; Dijkstra et al., 2006; Ronald et al., 2007).

However, these multi-scale models typically combine microscopic and macroscopic models in a very simplistic fashion. Thus the limitations of small-scale (e.g. being short-sighted) and large scale (e.g. not considering other moving pedestrians for route choice decisions) are usually not resolved. Information between the layers is not exchanged from the microscopic layer to the macroscopic layer and vice versa. Typically, the macroscopic layer provides the next intermediate destination to pedestrians steering in the microscopic layer and no information sharing between the microscopic and the macroscopic layer takes place.

The focus of the paper is the combination of advanced microscopic navigation strategies with macroscopic navigation strategies. Both state-of-the-art concepts alone including their qualitative properties, especially their realism with respect to real world scenarios, have been studied extensively in the literature. Combining both concepts, the realistic qualitative properties are inherited but the drawbacks of microscopic navigation concepts (extensive computation times) and macroscopic navigation strategies (difficulty to estimate travel times) are resolved.
We propose a new holistic multi-scale model unifying the advantages of the single layers and overcoming their limitations. Information is shared between the layers, such that the above mentioned issues are resolved. At the same time the approach allows an efficient realisation from a computational point of view. We demonstrate this new approach by extending a well established cellular automaton model (Köster et al., 2010). However, the concepts can be easily generalised to other microscopic pedestrian simulators. Due to computational efficiency as well as its high degree of realism, we believe that the developed concepts provide a valuable tool for optimizing transportation systems, buildings, as well as large infrastructures such as transportation hubs or even urban areas.

This article is structured as follows: First, we review existing multi-scale models in Section 2 and outline their limitations. Section 3 describes in detail the construction and implementation of each layer. In Section 4 the combination of these layers is explained. To show the improvements in the simulation results, various tests have been conducted and are presented in Section 5. A discussion concludes the article.

2. State of the art and limitations of multi-scale models

Multi-scale models typically consist of two layers: a small-scale layer modelling the navigation of pedestrians to a designated destination and a large-scale layer modelling strategic navigation, i.e. choosing different (intermediate) destinations.

On the small-scale layer, pedestrians are steered to a designated destination by forces similar to Newtonian particle mechanics. Typically, these forces are modelled using an attractive force or potential-based approach. Assuming a conservative force, i.e. the local force is given by the gradient of the potential, both approaches basically coincide. Continuous social force models (e.g. Helbing and Molnár, 1995; Lakoba et al., 2005; Löhner, 2010; Parisi et al., 2009; Chraibi et al., 2010, 2011a; Yu et al., 2005) typically use a force-based description whereas cellular automata, i.e. space-discretized models, typically rely on a potential-based description.
In the following, we will restrict ourselves to a potential-based cellular automaton model (Köster et al., 2010), which can be always interpreted in a force-based description assuming that forces are given by the gradients of the corresponding potentials. In the potential-based approach, a pedestrian's movement towards the destination is realised by a decreasing potential value towards the direction of the destination. Obstacles and other pedestrians walking in the vicinity of the pedestrian are considered via a repulsion force or repulsion potential (e.g. Kretz and Schreckenberg, 2006; Chraibi et al., 2011b; Köster et al., 2010) and thus added to the potential value. Typically, it is assumed that single pedestrians have a repelling potential of a certain width, which might also have a certain spatial structure depending on the direction of movement (Klüpfel, 2003; Köster et al., 2010). Using a cellular automaton, these pedestrians can be detected very easily in the nearby surrounding by simply checking neighbouring cells. The potential-based steering behaviour is often referred to as floor field or navigation field based behaviour and a variety of approaches to construct corresponding potential fields have been proposed (e.g. Schadschneider et al., 2009; Burstedde et al., 2001; Blue and Adler, 2001; Klüpfel, 2003; Köster et al., 2010; Nishinari et al., 2004; Kretz, 2009; Varas et al., 2007; Yamamoto et al., 2007; Guo et al., 2011).

To model the large-scale behaviour, i.e. the navigation strategy of pedestrians, graph-based approaches are used in most multi-scale models (e.g. Gloor et al., 2004; Gunnar G., 1998; Teknomo and Millonig, 2007; Kirik et al., 2009; Höcker et al., 2010). Based on a given geometry, possible routes are identified and from these a graph is constructed. According to given preferences, pedestrians move through the graph from their origin to their final destination via intermediate destinations as given by the vertices of the graph. Such intermediate destinations could be crossings or landmarks. The decision of which way to take at each intermediate destination might depend on environmental conditions, e.g. choosing illuminated paths in the evening, and can vary from pedestrian to pedestrian, e.g. take the shortest path, take the fastest path, avoid congestions, follow signage, follow friends, etc. as discussed in Pelechano and Malkawi (2008) or Golledge (1999). The different strategies are typically modelled by different routing algorithms. These reflect different behaviours by determining the best paths with respect to cer-

tain criteria, i.e. certain metrics or edge weights. However, most of the algorithms use travelling times as edge weights, which are often approximated by heuristics or mean values.

So far, these two layers have been combined to form a multi-scale model as follows: The navigation graph is used to generate pedestrians' paths based on a specific navigation strategy. Paths themselves consist of a list of intermediate destinations. The navigation field is then used to navigate pedestrians between these intermediate destinations until the final destination is reached. Although the combination of the layers already improves the realism of simulations, since small-scale aspects (e.g. avoiding other moving pedestrians in a close vicinity) and large scale aspects (e.g. navigation strategy) are addressed, there are still several open issues to be resolved:

On the small-scale layer, typically only static floor fields are considered (e.g. Burstedde et al., 2001; Blue and Adler, 2001; Klüpfel, 2003; Köster et al., 2010; Nishinari et al., 2004; Kretz, 2009; Varas et al., 2007; Yamamoto et al., 2007), i.e. floor fields determined in the initialization phase of the simulation. Other pedestrians who are not in close vicinity, e.g. large congestions, are not taken into account. Thus, such static navigation behaviour often leads to unrealistic simulation results: Pedestrians steer towards the congestions very closely until they "see" the congestion. In reality, pedestrians are able to see a congestion earlier if it is located in their field of vision. Only recently two approaches have been proposed independently to include dynamic aspects (Kretz, 2009; Hartmann, 2010). This however requires a continuous update of the navigation fields, which is computationally expensive, particularly for large domains. Besides, using these dynamic floor fields, pedestrians consider congestions, which are not visible from their actual position while making their navigation decisions. An example is given in Figure 1.

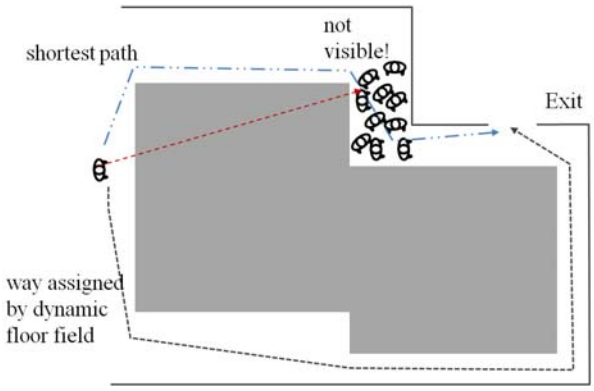


Figure 1: Navigating pedestrians on a dynamic navigation field, the pedestrian on the left side would be routed around the south corner of the obstacle due to the congestion in the upper west corner. Since the pedestrian is not able to see the congestion, he would probably choose the other, shorter route in reality.

On the large-scale, routes that are based on heuristic edge weights are typically determined, in most cases heuristic estimates of travelling times. These are usually estimated by the number of pedestrians moving along an edge (i.e. edge densities, e.g. Kirik.2009). However, since densities are local properties (e.g. congestions) and are not constant on the whole edge area, these estimates are not sufficiently exact. Consider for example a long edge, where a small congestion occurs at the very end (c.f. Figure 5). This edge would not been chosen due to the longer travel time derived from the slow velocity within the congested area, although the congestion would vanish until a pedestrian reaches this location.

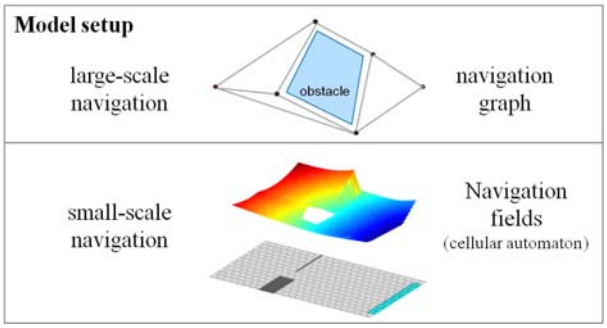


Figure 2: Schematic illustration of the different implementation layers

3. Ingredients of a holistic multi-scale model

The implemented multi-scale simulation model consists of two layers: A cellular automaton for discretization of space and time, which serves as the underlying grid for constructing the navigation field, and a navigation graph to model the large-scale navigation behaviour. Figure 2 illustrates the setup of the model. In the following, each layer is described in detail.

3.1. Cellular automaton with navigation fields

For the discretization of space and time we apply the cellular automaton introduced in Köster et al. (2010). This cellular automaton consists of hexagonal cells where each cell has the size of an average European male according to Weidmann (1993). Thus, exactly one person fits into one cell at each time step. Obstacles, origins and destinations are discretized accordingly.

The movement of each pedestrian is influenced by different forces, modelled via potentials: Repellent forces of obstacles and other moving pedestrians as well as a driving force to the destination. These forces are represented by a navigation field (or floor field). In each update step, pedestrians who are allowed to move (not all pedestrians are allowed to move in each time step to realize different speeds) search for an accessible neighbouring cell with the lowest navigation field value. All pedestrians are updated sequentially (first born - first updated).

These navigation fields are implemented as scalar fields $T(\vec{x})$, which are increasing monotonically with an increasing distance from the destination to which the pedestrians are steering. In Figure 3, a scenario and the corresponding navigation field is illustrated. Such a field can be constructed using different approaches (Dijkstra, 1959; Nishinari et al., 2004; Hartmann, 2010).

Here, we will make use of the approach introduced by Hartmann (2010). The central idea is based on the analogy of an expanding wave. It is expanding from the destination normal to the wave front with a certain velocity, e.g. depending on the local pedestrian density. Arrival times of the wave are interpreted as estimated travel times for pedestrians. These can be used as navigation fields since moving down the spatial gradient of this estimated travel time field implies the minimization of travel times - a natural human behaviour.

The wave propagation is described by the Eikonal equation (with speed $F(\vec{x}) \geq 0$ in the normal direction):

$$\begin{aligned} F(\vec{x})|\nabla T(\vec{x})| &= 1 \text{ in } \Omega, \\ T(\vec{x}) &= 0 \text{ in } \Gamma, \end{aligned} \tag{EIK}$$

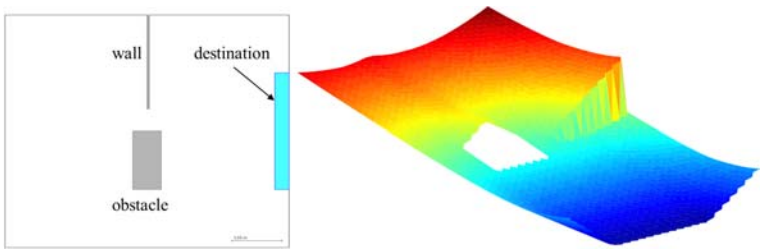


Figure 3: Left: Scenario, Right: Corresponding field; The shape of the obstacle inside the field is distorted due to the perspective error of the 3D-screenshot. The imbricated border around the obstacle results from the underlying hexagonal grid.

where $T(\vec{x})$ is the arrival time of the wave, which originated at $T = 0$ from the destination $\Gamma \subset \Omega$. Choosing $F(\vec{x}) \equiv F$ constant in the whole domain Ω except in obstacles ($F(\vec{x}) = 0$), arrival times $T(\vec{x})$ correspond to travelling times of a pedestrian moving with speed F from Γ to $\vec{x}$ along the shortest and thus fastest path (Huygens principle, Kretz, 2009). And vice versa, the times $T(\vec{x})$ correspond to expected travelling times of pedestrians travelling backwards, i.e. with speed F from $\vec{x}$ to the destination Γ. By construction, the direction of the fastest path from any point $\vec{x} \in \Omega$ to Γ is given by the gradient of $T(\vec{x})$ (Hartmann, 2010). The Eikonal equation (EIK) can be solved efficiently using the Fast Marching Method (FMM, Sethian, 1999). The FMM, which is basically a modified heap sort algorithm with complexity $O(N \log N)$, is outlined in Appendix A for completeness. The computational efficiency of the FMM can be further increased by not calculating (EIK) on the whole domain Ω, but rather stop the calculation if a corresponding navigation field has been calculated for all positions, i.e. cells, occupied by pedestrians moving to the destination Γ.

Modelling the wave expansion using a constant speed $F(\vec{x})$, other pedestrians are not considered in this navigation step and the navigation field only has to be calculated once. However, to consider other pedestrians during the calculation of the navigation field, the realism of the navigation is increased. As a consequence, the field has to be updated / recalculated dynamically. In Figure 4, the solution of (EIK) with $F(\vec{x})$ scaling inversely with the local pedestrian density, is shown schematically. To be more precise, we choose $F(\vec{x}) = 1$ if a cell of the cellular automaton is not occupied and $F(\vec{x}) = 1/(1 + \delta)$ with a penalty factor $\delta > 0$ if a cell is occupied by a pedestrian. One clearly observes that the wave originating in the destination V^D is propagating faster around the group of pedestrians. Since at any point the direction of the fastest path is given by $\nabla T(\vec{x})$, the fastest path from origin V^O towards the destination V^D can be determined explicitly as shown in Figure 4. The time estimate following the fastest path is given by $T(\vec{x})$. Estimating travel times from the Eikonal equation assumes that pedestrians always follow the shortest path according to the current situation. But since the situation is evolving, i.e. pedestrian densities change over time, it can be only an estimate. The accuracy depends on how dynamically pedestrian densities change. If they do not change, i.e. pedestrian densities are quasi-static, the Eikonal equation would estimate travelling times for the fastest path up to numerical accuracies of the underlying discretization. However, in reality as well as in the simulation, there are no purely rational pedestrians following fastest paths, e.g. pedestrian movement in the simulation is always subject to small stochastic effects. Therefore, the Eikonal equation can give only estimates.

Although the dynamic approach of Hartmann (2010) solves the unrealistic behaviour of simulated pedestrians on a small scale, fastest paths are estimated too optimistically with such dy-

namic fields on a large-scale: Pedestrians would steer around congestions, which are possibly not even visible from their actual position (c.f. Figure 1). Thus, a distinction between visible and invisible areas is mandatory. Here, the navigation graph from the large-scale layer is applied, as described in the next section. Additionally, the combination with a navigation graph allows one to efficiently control the computational complexity of the construction of navigation fields and thus reducing the overall run time of the simulation significantly. A detailed description of how this coupling with a visibility graph increases realism is given in Section 4.

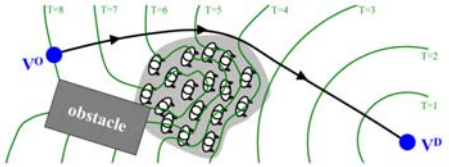


Figure 4: Schematic sketch of a dynamic navigation field estimating travel times of pedestrians travelling form the origin V^O to the destination V^D. The direct way is blocked by a group of pedestrians through which one would move slower than on free fields. Thus the fastest path is going around the group.

3.2. Navigation graph

To take large-scale aspects like different navigation behaviours into account, a navigation graph is constructed on top of the scenarios' geometry. The graph can be created manually or derived automatically. The latter approach is described in Appendix B. For a detailed description, we refer to Kneidl et al. (2011).

By applying different routing algorithms on the graph, different wayfinding behaviours of pedestrians can be modelled. To be able to distinguish different route choices, e.g. the shortest path, the fastest path, closest path along the airline, following signage, or avoiding congestions, different metrics are applied. Although each metric is defined slightly differently, all rely on the definition of edge weights. The easiest and most straight forward idea is to derive the edge weight from the distance between both vertices of an edge. Since here we are solely applying visibility graphs, the Euclidean distance can be used as the edge weight. However, by adopting this approach, we must consider that pedestrians already moving on an edge can cause congestions. Thus travel times, i.e. a combination of distance and velocity, seem to be a more natural edge weight. Since the velocity is changing depending on the local density (Weidmann, 1993), densities on the edges have to be determined. On the cellular automaton level, local densities are given by the number of pedestrians that are located in neighbouring cells of a single pedestrian. On the graph layer, local density can only be measured indirectly e.g. by the maximum of the local densities in the vicinity of pedestrians travelling along that edge. Using this approach, the number of pedestrians moving along an edge is determined indirectly, thus densities and travel times are only rough estimates and not precise enough for an exact measure. An example for the roughness of the estimates is given in Figure 5. Here, a very long edge is considered with a congestion in front of a bottleneck at the very end of the edge. If maximum densities are taken in order to derive travel times, this edge would be assigned a very long travel time, although the congestion does not affect the pedestrian who wants to enter the edge, since it is rather small.

In addition to the possibility of realising different navigation strategies, the navigation graph enables dynamical changes to the simulated scenarios during runtime. This allows us to incorporate events, such as streets blocked by congestions, fire (Mayer et al., 2011) or emergency

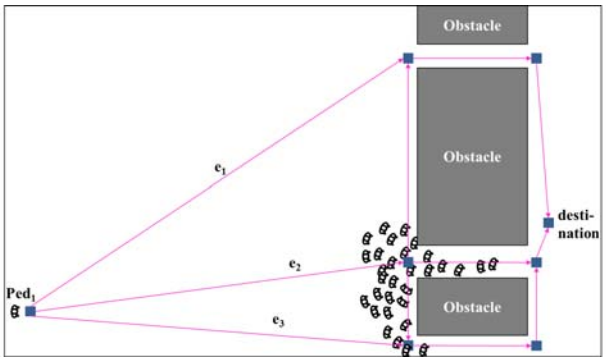


Figure 5: Example for travel times which are derived from maximum densities: Edge e_2 would not be chosen because of the small velocities in front of the bottleneck, although the congestion would be resolved when pedestrian ped_1 reaches the bottleneck.

vehicles. To alter a scenario during runtime, the graph can be used very efficiently, by either deleting the corresponding edge or by assigning a very high edge weight. The detection of such an event on an edge can be performed in different ways: Either by definition upfront (at a certain time step a specific edge is not accessible any more), by dynamic detection (if congestion occur, this edge is not accessible any more and therefore should not be considered any more) or by the user, who interactively defines an event.

Although graph-based approaches allow a large flexibility with respect to macroscopic navigation behaviours, the efficient estimation of realistic travelling times is an open problem. The realism of the macroscopic navigation behaviour and thus of the simulation depends on these estimates. The holistic multi-scale approach outlined in the following section significantly improves the estimation of travelling times and thus the predictive power of a simulation approach.

4. Combining the layers to a holistic multi-scale model

These individual implementation layers are combined to use synergy effects: The navigation graph divides the space into visible and invisible areas for each pedestrian's location. The vertices of the navigation graph are used to calculate smaller navigation fields while values derived from these navigation fields can be used as input for travelling time estimation, i.e. deriving edge weights. The advantage is twofold: on the one hand, estimation of travelling times on the navigation graph layer is improved and on the other a significantly more efficient calculation of the navigation field is possible. The latter is important with respect to a possible application in an interactive simulation software for virtual training of security staff, since it has a major impact on the overall runtime.

As described in the previous section, a dynamic recalculation of the navigation field is necessary to get a realistic walking behaviour of the pedestrians (Kretz, 2009) (c.f. Figure 4). However, recalculation of the whole navigation field will result in recalculating the values of many cells which are not affected at all, i.e. areas where no pedestrians move. Using the navigation graph, the area of recalculation can be kept small by dividing the whole navigation field into small navigation fields, each defined between two adjacent vertices of the graph. Now, having these small fields, only those fields where pedestrians walk have to be created. And among these only those have to be dynamically updated in regions where pedestrians with a high local density walk, e.g.

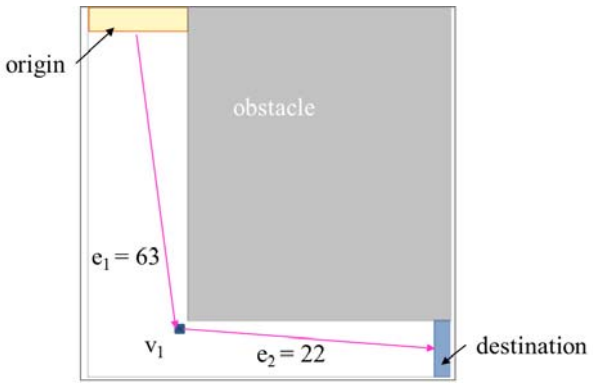


Figure 6: Scenario for testing the impact of dynamic navigation fields: Pedestrians walk from the origin to the destination around the corner. The graph is depicted in pink. The number at each edge describes the number of recalculations of the navigation field.

the number of pedestrians per edge or the local density of pedestrians moving along a certain edge exceeds a certain threshold.

Using the navigation graph, a second advantage comes into play. Since the graph covers all the possible ways the pedestrians can walk on, we do not need to calculate the navigation field in an undirected fashion, i.e. in all directions from the destination. The graph allows us to construct directed navigation fields between two adjacent vertices in the macroscopic navigation graph (in the following these two vertices will be denoted as V^O and V^D and pedestrians are assumed to move from V^O towards V^D): Following the ideas of Petres et al. (2007) considering the navigation of single underwater vehicles, a directed navigation field can be created. The main idea is to sort the vertices $\vec{x}_i$ of the computational grid in the FMM-Algorithm (c.f. Appendix A) not based on the key $\kappa(\vec{x}_i) = T(\vec{x}_i)$, but rather on a combination of smallest time $T(\vec{x}_i)$ and the distance $d(\vec{x}_i, V^O)$ to the origin of the pedestrians, the vertex V^O:

$$\kappa(\vec{x}_i) = \alpha \cdot T(\vec{x}_i) + (1 - \alpha) \cdot \beta \cdot d(\vec{x}_i, V^O),$$

where $0 \leq \alpha \leq 1$ and $\beta > 0$ are appropriate constants. This enforces a preferable update of cells closer to the starting point V^O of pedestrians, i.e. a more or less directed calculation (depending on the choice of α and β) of the navigation field from the (intermediate) destination vertex V^D to the (intermediate) origin vertex V^O. The concept of a directed calculation ensures that a navigation field is calculated in a minimal area and thus also ensures an optimal computational complexity. A comparison between an undirected navigation field and a directed navigation field with different values for α for an edge e_1 of a sample scenario (illustrated in Figure 6) is given in Figure 7. The undirected navigation field propagates in all directions, which leads to a very large field, where many of the cells covered are not geometrically closely located to the edge. The figure on the right-hand side shows the results of different values of α: The influence of α and β on the shape of the resulting field is huge, i.e. for which cells the field is calculated. However, it has only a very small influence on the actual values of the navigation field, i.e. it has no influence on the quality of the simulation results. The values are typically predefined by an experienced user by testing different combinations of the specific scenario to be able to adapt the calculation effort (depending on the size of the scenario).

Another advantage with respect to the graph layer follows directly: In previous works we

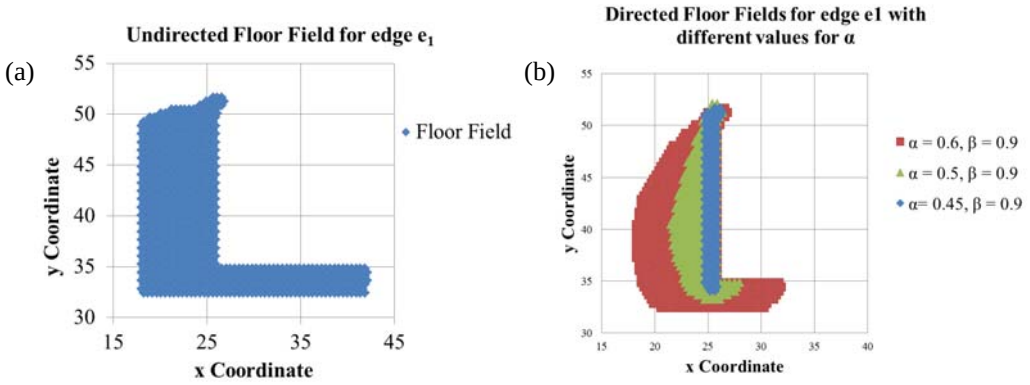


Figure 7: Illustration of an undirected field compared with directed fields, using different α values for edge e_1. The wave propagates from vertex v_1 towards the origin (c.f. Figure 6). The undirected navigation field propagates in all directions, which leads to a very large field, where many of the cells covered are not located geometrically close to the edge.

have assumed a constant velocity on an edge, depending on the maximum density measured on each edge to estimate travelling times used as edge weights (c.f. Figure 5). But generally, the speed of pedestrians is a local (in time and space) property and not a constant, e.g. pedestrians are moving slower uphill or in dense crowds.

In our approach, the estimation of travelling times from V^O to V^D, obtained by the solution of the Eikonal equation, is used to estimate travelling times for corresponding navigation algorithms of pedestrians on the graph. Adopting travelling times $F(\vec{x})$ in equation (EIK) depending on the local terrain and pedestrian density resolves the problem of estimating travelling times for navigation graphs.

This is a very elegant way to estimate travelling times within dense crowds, since travelling times, which are already calculated by the dynamic navigation field, are used. These values are available from the construction of the dynamic navigation field and it is not necessary to estimate the travelling times additionally: The distance, i.e. travelling time, can be assigned directly by taking the value of the dynamic navigation field of the destination cell.

Additionally, the issue of being too ”intelligent” (in the sense of having global knowledge about the current state of the entire scenario) by navigating on a dynamic navigation field can be resolved (c.f. Figure 1). For calculating a route for a pedestrian, two sets of edges have to be distinguished for each pedestrian: visible edges and invisible edges. For visible edges, the navigation field values are taken, whereas for invisible edges only mean travelling times are considered. This improves the modelling of pedestrian behaviour, since pedestrians are not able to look around corners.

Furthermore, it is possible to minimise computational effort by continuously updating the navigation field / expected travelling times on an edge only if the number of pedestrians travelling on that edge or the corresponding densities, i.e. the local densities of pedestrians corresponding to that edge, exceeds a certain threshold. If the pedestrian density on the edge is small, one does not expect a significant impact on travelling times, thus this threshold criterion still yields appropriate estimates of travelling times while simultaneously reducing computational efforts significantly.

By combining the layers (c.f. Figure 8), we have solved several issues: First of all, using

the values of the navigation field within the navigation graph leads to a precise definition of edge weights for visible edges, which we believe comes very close to reality. Secondly, by only having small areas that need to be updated continuously and in addition using a directed navigation field, the computational effort is minimised. This allows to rely on a continuous update of floor fields. Thus not only the travelling time estimation of the graph layer is more exact, but also the small-scale behaviour is improved. Neither movement patterns like walking in single file occur nor simulated pedestrians are short-sighted anymore as they have been when using only static fields without considering any other moving pedestrians.

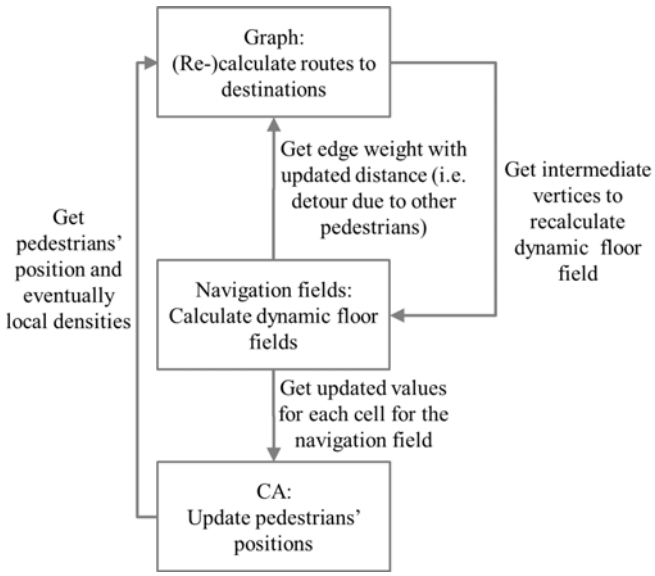


Figure 8: Interaction between the layers of the holistic simulation model

In the next section, simulations of various examples are given to illustrate and underline the improved realism as well as computational efficiency of the presented new holistic multi-scale approach.

5. Tests

To demonstrate the improvement of the simulation results, we have conducted a series of tests. These include basic tests as well as an elaborated realistic test case considering pedestrian flows in the central city area of Munich, Germany.

For all screenshots, following color code is used: Green dots represent pedestrians who can move freely, yellow dots depict denser areas, dots in Red refer to pedestrians in very dense environments, who are not able to move since neighbouring cells are occupied.

5.1. Basic tests

The first test case simulates pedestrians walk around a corner. The scenario is illustrated in Figure 6. In total, 1,200 pedestrians are simulated, with a rate of 6 pedestrians per second leaving the origin. The results are summarised in Figure 9.

Without using a dynamic field, pedestrians walk very closely around the corner and do not use the available space of the corridor. One can clearly observe that with the static field, a congestion occurs in front of the corner. It gets even worse with more pedestrians that enter the scenario. Using a dynamic floor field with different values $F(\vec{x})$ (c.f. equation (EIK)) on those cells which are occupied by pedestrians, the pedestrians spread wider and make use of the whole corridor. Here, the values $F(\vec{x})$ depend on the surrounding cells. The more neighbouring cells of a single cell that are occupied, the higher the value. Pedestrians do not try to walk along the shortest path (at least not until they have passed the congested area), but choose a path avoiding the congestion, i.e. the fastest path. This behaviour is observable in both cases, using the navigation graph and using the navigation graph. Thus, no congestions occur in front of the corner, which seems more natural referring to Kretz (2009).

To test the impact on computational times of using intermediate destinations for defining specific navigation fields, the same example is used: We simulate in total 1,200 pedestrians, six pedestrians leaving the origin per second. The overall simulation time is 600 seconds, as in the experiment before. The same three different test cases are defined: First, the simulation is performed without dynamic fields to retrieve the benchmark value. Secondly, the simulation is performed using undirected dynamic navigation fields. The update interval of the navigation fields is every 10 microsteps (= 1.7 seconds). The third setup uses dynamic navigation fields in combination with the navigation graph. The navigation fields are constructed using a directed search and the update interval is defined by the relative change of local density on the edge: If the absolute value of the maximum local density (number of neighbour cells that are occupied by a pedestrians divided by the number of accessible neighbouring cells) on an edge changes for more than 10 percent, the field is recalculated. For edges, which are not used, no navigation field is created. In Figure 6 the number of recalculations per edge is depicted. In Figure 10 the qualitative measures are shown: The time of a simulation run with a static navigation field serves as a benchmark, and the run times of simulations with dynamic fields are compared to this benchmark value. By using a navigation graph, i.e. using directed navigation fields, the run time does not increase. Without using a graph, i.e. in the undirected and not partitioned calculation of navigation fields (no intermediate destinations are given without a graph, thus the whole scenario has to be updated), the runtime increases significantly even for this small scenario.

5.2. Test for visible vs. invisible edges

To show the effect of distinguishing between visible and invisible edges, a second test is conducted. As stated in Section 2, pedestrians can only take congestions into account for choosing alternative routes, which are visible to them. To show the impact of distinguishing these edges, a test scenario is simulated once without a navigation graph (no distinction) and once with a navigation graph (distinction possible). Pedestrians walk from the west side of the scenario (c.f. Figure 11) towards a destination, which is located on the eastern part of the scenario. The direct path is obstructed by different obstacles and a bottleneck, which is not visible from the origin. Thus, it is expected that pedestrians first walk around the northern corner, and once the bottleneck is congested, walk around the south corner of the lower obstacle of the bottleneck. The results of the simulation without the usage of a navigation graph show a different result: Since a globally shortest path is always found, pedestrians are routed around the south corner of the first obstacle to avoid the bottleneck. This seems not likely to be a natural behaviour, since the pedestrians cannot see the congestion before passing the first obstacle. By using a graph and thus being able to distinguish between visible and invisible edges, the simulation results look as expected: All pedestrians pass the first obstacle around the upper corner and start - after the bottleneck gets

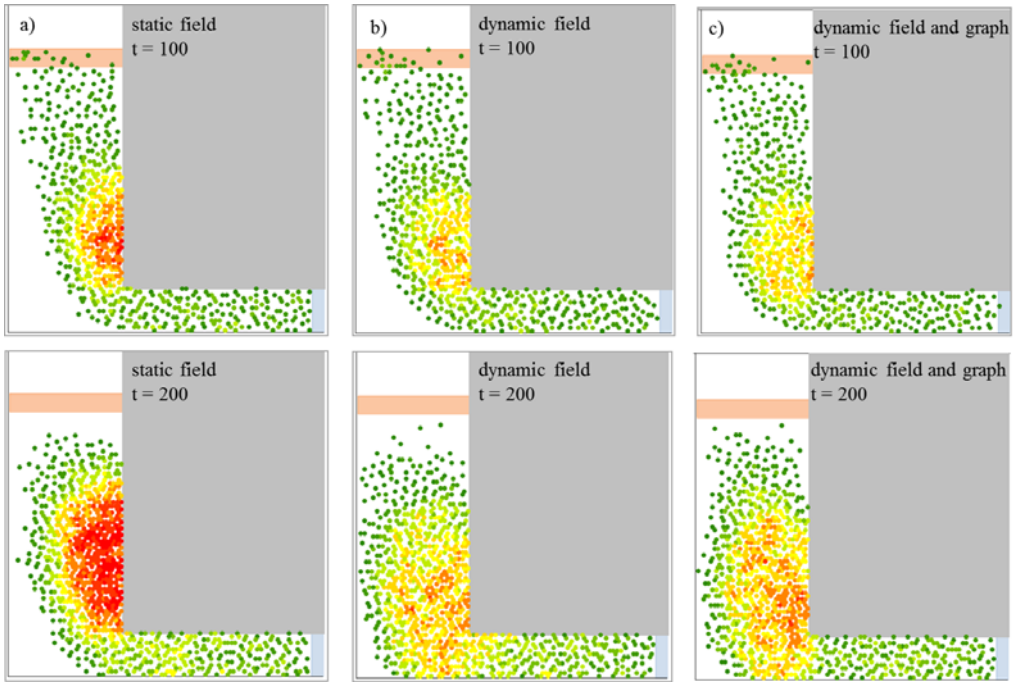


Figure 9: Simulation screenshots after 100 and 200 seconds (a) snapshot with static navigation field; (b) snapshot with a undirected dynamic navigation field; (c) snapshot with directed dynamic navigation fields in combination with a navigation graph

Simulation scheme	Relative time factor
Static navigation field	1.00
Dynamic navigation field without navigation graph	6.96
Dynamic navigation field with navigation graph	1.10

Figure 10: Comparison of the computational efficiency of simulations using static navigation fields as well as dynamic navigation fields with and without the use of a navigation graph. Relative time factors with respect to the static navigation field simulation without a navigation graph are shown.

congested - to walk around the southern corner of the lower obstacle of the bottleneck. In Figure 11, the traces of all simulated pedestrians are shown as well as snapshots of each simulation run to show the densities.

5.3. Munich city centre scenario

Beside the simplistic technical tests presented above, we further consider a more realistic setup, namely pedestrians walking in the city centre of Munich (Germany) to a specific destination. The scenario consists of a 2D-map of Munich, Germany, consisting of all obstacles and accessible areas, i.e. streets and places. The simulated area size is 731 $m \times 545$ m and the detailed geometry of the scenario is taken from Open Street Map (www.openstreetmap.org). For simplicity, we do not separate between streets and pavements. Likewise, traffic lights have not been considered. In total, we simulate 150 pedestrians. Pedestrians start walking from the south west corner of the scenario to a place in the north east corner - the Marienhof.

In the first test case, the simulation is based solely on a static navigation field without a navigation graph. The results of all pedestrians' traces are shown in Figure 12. It can be clearly observed that only one path is chosen by all pedestrians. This quite unrealistic result is due to the fact that there is no large scale layer which models orientation. Thus, every pedestrian is walking along the gradient of the navigation field, i.e. walking on the fastest path.

The second test case makes use of a navigation graph and different navigation algorithms: A percentage of the pedestrians move along the fastest path, using the navigation field values at intermediate destinations as edge weights. Another portion of the pedestrians follows a path, which leads as close as possible along the airline to the destination. These paths are found using the A* Algorithm, which is based on two different types of weights for the search within a graph: A fixed (or known) part, namely the length (or time) of the already traversed edges, and an estimated value, which maps the unknown part of the remaining path. The first type of weight refers to the values of the navigation floor field as used for the fastest path algorithm, the second type refers to the (remaining) air line distance to the destination (Höcker et al., 2010; Kneidl and Borrmann, 2011).

A third type of pedestrian is modelled with the Straight and Long Leg (SALL) algorithm: According to Golledge (1999) and Kneidl and Borrmann (2011), pedestrians tend to prefer long and straight legs on their path to a destination in an urban environment. The algorithm models this behaviour explicitly: it takes into account the directional changes at intermediate destinations, i.e. the angle between the incoming edge and possible outgoing edges. The second aspect of

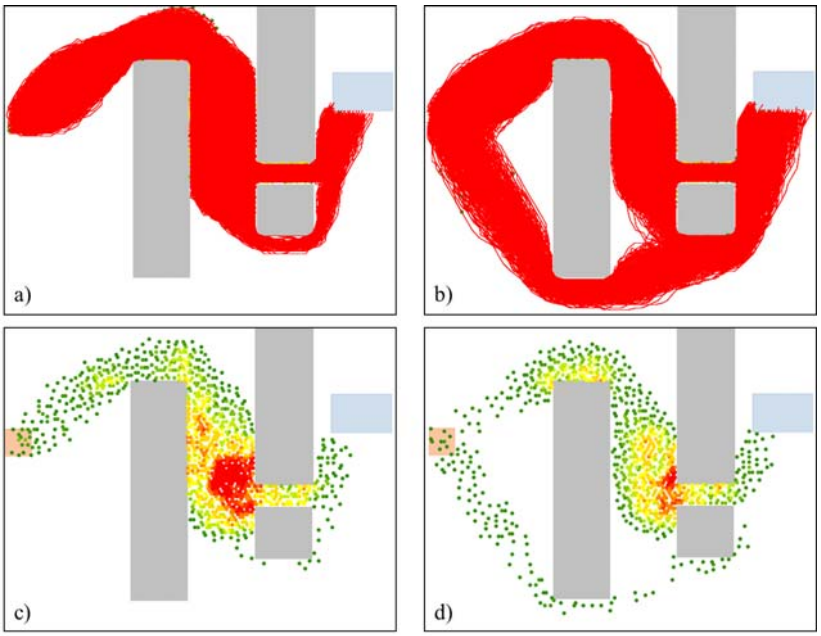
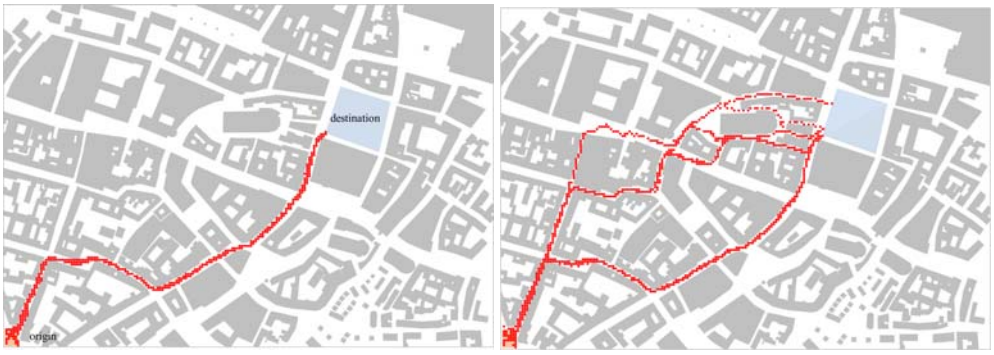


Figure 11: Simulation results: (a) traces of all pedestrians for the simulation with using a graph; (b) traces of all pedestrians for the simulation without graph usage; (c) simulation screenshot of the simulation with graph; (d) simulation screenshot for the simulation without navigation graph



(a) (b)

Figure 12: (a) Mainstream of pedestrians, walking from south west towards the destination in the north west corner. The scenario is simulated without using a navigation graph. (b) Mainstream of pedestrians, walking from the south west corner towards the destination in the north west corner. The simulation uses a navigation graph and three different routing algorithms to model different navigation behaviour.

(a) (b)

Figure 13: (a) Screenshot of a simulation using a static field. (b) Screenshot of a simulation using a dynamic navigation field in combination with the navigation graph.

the algorithm is the length of one leg, i.e. how long can a pedestrian walk along one direction before he has to make a turn. A third part of the weight is the relative distance improvement to the destination by choosing the corresponding edge.

Simulation results based on a combination of all three types within one simulation are shown in Figure 12. As one can observe, different routes have been chosen by the three types of pedestrians. The fastest path conforms with the simulation without the navigation graph; the two other algorithms model pedestrians without knowledge of a location respectively finding their way according to different criteria.

Comparing the runtime of simulation runs, the graph-extended simulation is slower than the simulation with the static navigation field by a factor of approximately 1.3. With the improvement of the results in mind, i.e. a significantly better route choice, this factor is well acceptable.

To show the improved walking behaviour by using the dynamic navigation field on a larger scale, the Munich City Scenario is extended to a total of four origins and four destinations. Each destination is reachable from each origin. Pedestrians are equally distributed to the different destinations.

Two simulation runs are conducted: First, without using dynamic navigation fields, i.e. using only static fields (referred to as static simulation), and secondly using the dynamic fields in combination with the navigation graph (referred to as dynamic simulation). To get a better comparison, all pedestrians follow the same routing strategy, walking along the fastest path in the dynamic simulation. In the static case, pedestrians walk along the fastest (i.e. shortest) path by definition. In each origin, 600 pedestrians are generated, such that a total of 2,400 pedestrians are simulated. In the dynamic simulation, a new navigation field is recalculated if at least 60 percent of the surrounding cells of a pedestrian walking towards the corresponding destination is occupied. The total simulation time is 600 seconds, and within that time, 90 recalculations of highly frequented edges (in total 16) are necessary.

Screenshots of the simulation at the time step, when half of the pedestrians have left the origins, are shown in Figure 13. It is clearly observable (e.g. in the east part) that by using the dynamic field pedestrians use more space and do not walk in single file like they do in the simulation with the static field. This is explained by the fact that using a static field, other pedestrians are recognised only in the radius of approximately one meter. Thus, pedestrians

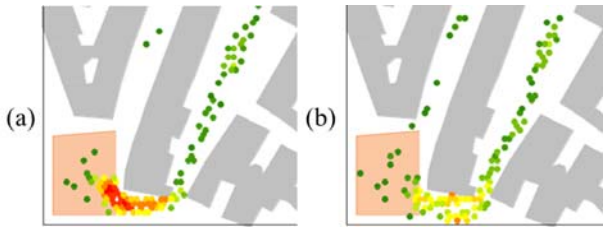


Figure 14: Zoom into the lower west corner of the scenario: (a) static simulation; (b) dynamic simulation

are ”short-sighted” and higher occurring densities are not recognised early enough. Secondly, one can observe the re-routing of the pedestrians due to the use of a dynamic navigation graph (e.g. in the south-west corner of the scenario). To analyse the walking behaviour in more detail, Figure 14 depicts a ”zoom” into the lower west corner of the scenario. In the static case, a dense area evolves, since the shortest path to the eastern destinations is along the east leg leaving the origin. These high densities are again explained by the fact that the static navigation field does not consider pedestrians for routing decisions that are further away than one meter. Thus, only if a pedestrian comes very close to the congested area, he realises the congestion and tries to evade it. In the dynamic case, pedestrians are considered with a certain penalty value directly within the navigation field via $F(\vec{x})$. Thus, no congestion occurs, since the western route is noticed to be faster after a short time, if many pedestrians are blocking the more direct way. This clearly mimics real world behaviour, pedestrians would not walk inside dense areas if there is a possibility to move around. Although not proven empirically, the results of the dynamic simulation seems to lead to a more natural behaviour of pedestrians.

To summarise, all tests have backed up our assumptions about the qualitative improvement of simulation results using a multi-scale approach. At the same time, the runtime of the holistic model increases only slightly by an acceptable factor.

6. Discussion

In this contribution, a new holisitc multi-scale approach for modelling pedestrian dynamics has been presented. Modelling the complex behaviour of individuals, the exchange of information between different levels of details retrieve the most realistic results for pedestrian pedestrian simulations. Each of the models on each layer is well established on its own, still a combination can improve the overall results in taking aspects on different layers into account.
On the space and time discretization layer of a simulated scenario (using a cellular automaton), pedestrians are moved on the grid (if possible) from one cell to the next in each time step. Navigation fields, which are created using the Fast Marching Method (FMM), are considered to navigate along the fastest way between an origin and a given destination. Moving pedestrians can be taken into account by dynamically updating the navigation field and mapping pedestrians occupying a cell to the metric underlying the FMM. This leads to significant improvements in the walking behaviour of pedestrians, as other pedestrians in sight and not only those in very close proximity are considered. The drawback of dynamic fields is that the whole field is updated and therefore - depending on a position of a single pedestrians - invisible areas are updated as well. This implies that congestions are detected which are not visible in reality. Although - in combination with a cellular automaton - the FMM is very efficient from a computational point

of view, with growing scenarios, the number of cells rises and thus leads to a poor performance. Considering a dynamic update, this performance drawback is even more severe.

However, many of the cells for which a dynamic navigation field is calculated, are not affected at all, i.e. pedestrians are walking only in a restricted area and in many regions the density is so small that the navigation field is effectively unaffected during a dynamic update. With our approach, we combine these navigation fields with a navigation graph - obtaining a multi-scale model. This navigation graph does not only serve as a basis for different navigational behaviours of pedestrians, e.g. following the shortest path, the fastest path, signage, or avoiding congestions, but it helps to distinguish between visible and invisible areas for each single pedestrian and thus improves the navigation.

The interaction of the two layers allows to use synergy effects: Using intermediate destinations of the navigation graph, we can minimise the area that has to be recalculated dynamically in order to keep the dynamic navigation field updated. Furthermore, a directed search can be conducted by sorting the cells relative to the smallest time and distance to the origin rather than the arrival times of the wave as in the original FMM. By initialising a recalculation of floor fields via directed FMM only if a threshold density is reached and stopping once a navigation field has been calculated for all cells occupied by pedestrians, the computational efficiency can be further increased. The graph on the other hand can use the derived values of the navigation fields to assign travel times as edge weights. Using these values leads to a robust measure of travel times, since local congestions and resulting detours are taken into account.

The results presented in this work support the expected theoretical considerations. By realising the combination of the two layers, the simulation has improved significantly. Not only the walking behaviour of pedestrians has been improved, i.e. the effect of walking in single file is resolved as well as a more natural route choice due to distinction between visible and invisible edges, but also computational effort can be minimised by defining smaller navigation fields, respectively. The overall computation time compared to classical approaches using static floor field based navigation is increased only mildly.

This model therefore provides an extremely realistic and fast computational method to simulate large pedestrian crowds in real-time. However, effort still has to be made to further improve the details of the model. As we are using a graph, pedestrians can be only re-routed at intermediate destinations. This is a limitation, as the intermediate destination is sometimes congested. A definition of different criteria to re-route may be one possibility. Another open point is the shape of the intermediate destination. Which shape works best for which scenario? Secondly, validation of the model is an ongoing task. The different route choice algorithms have to be tested on different set-ups (e.g. urban environment, public buildings, large events) and calibrated, respectively. Unfortunately not sufficient real data exists to push validation further, at least not so far.

Appendix A. Fast Marching Method

Given a discretization of Ω, e.g. the dual grid of cells used in microscopic pedestrian simulators based on cellular automata, the Fast Marching Method (FMM, Sethian, 1999) offers an efficient technique so solve the Eikonal equation (EIK). Let us shortly recapitulate the main idea of the FMM, which is based on the concept of causality. Vertexes $\vec{x}_i$ of the computational grid underlying the FMM are grouped in three states: far away (no $T(\vec{x}_i)$ assigned), trial ($T(\vec{x}_i)$ assigned but not fixed) and fixed ($T(\vec{x}_i)$ assigned and fix). Starting with a discretization of the initial front, the trial vertex $\vec{x}_i$ of the computational grid with the smallest time $T(\vec{x}_i)$ is chosen

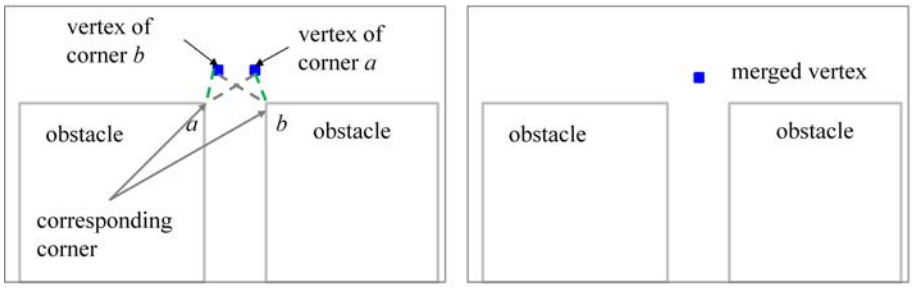


Figure B.15: Example for two geometrically close vertices that can be merged, as they refer to the same corner points

and set to fixed. Its trial neighbours are updated, i.e. T is recalculated, and its further away neighbours are initialised, i.e. T is calculated, and moved to the trial set. These steps are then repeated for the trial vertex with the smallest T. The algorithms stops if the trial set is empty or if appropriate stopping criteria are reached, e.g. for all cells $\vec{x}_i$ occupied by a pedestrian a $T(\vec{x}_i)$, i.e. a navigation field, has been calculated. Thus, the FMM is basically nothing else other than a classical sorting algorithm iteratively finding the trial vertex with the smallest time T, moving this vertex to the set of fixed vertexes and updating its neighbours. Thus adapting an efficient implementation of a sorting algorithm, e.g. a heap sort, the FMM has an optimal computational complexity of $N \log N$ (Sethian, 1999).

Appendix B. Automatic construction of large-scale graphs

The automatic construction of navigation graphs consist of two steps. Firstly, appropriate nodes are determined and secondly these are linked via edges. For completeness the construction is summarised and for more details we refer to Kneidl et al. (2011).

Appendix B.1. Automatic derivation of a navigation graph

To automatically derive a navigation graph form a given geometry, first navigation points are placed. These points refer to vertices within the resulting graph. They are placed at the bisector of every convex corner of each obstacle with a certain distance to the corner.

Furthermore, it has to be assured, that each vertex is placed in sight to the corresponding corner. Two adjacent vertices are merged into one new vertex, if they are (a) geometrically close to each other and (b) refer to the same corner points. Each vertex refers to at least one corner point, namely the point it was constructed from. If two obstacles are located very closely to each other, it may occur, that a corner of another obstacle is closer than the original corner. Thus, in such cases, two vertices can be merged, if both refer to the same corner points. Figure B.15 illustrates such an example.

For a detailed description of the navigation point placement, please refer to Kneidl et al. (2011).

Appendix B.2. Edge generation

To enhance the efficiency of later applied routing algorithms, one criteria of a connecting edge is visibility. The easiest way to connect two nodes is to check for visibility, and then to connect the two nodes, if this criterion is fulfilled. However, this would result in a very dense

graph. A dense graph would lead to the poor performance of our applied algorithms, especially if edges are supercilious. We define supercilious edges as geometrically close edges, i.e. outgoing edges which enclose a very small angle. Edges, which do not lead towards the destination, are considered supercilious as well.

Therefore, the following criteria are introduced: The angle between two outgoing edges can guarantee that no geometrically close edges are inserted. By choosing the angle size, the density of a graph can be controlled. Keil and Gutwin (1992) describe such graphs as Fixed-Angle-θ graph. The smaller the angle, the denser is the resulting graph.

To avoid edges that are not leading towards the destination, a directed search is conducted. By applying a spatial index that stores all geometric elements of a scenario, a nearest-neighbour range search is used to find vertices in direction to the destination. The details of the search technique can be found in Beckmann et al. (1990). The description of the whole algorithm, called cone-based search method, can be found in Kneidl et al. (2011).

Antonini, G., Bierlaire, M., Weber, M., 2006. Discrete choice models of pedestrian walking behavior. Transportation Research Part B: Methodological 40 (8), 667–687.

Asano, M., Iryo, T., Kuwahara, M., 2010. Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour. Transportation Research Part C: Emerging Technologies 18 (6), 842–855.

Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B., 1990. The R*-tree: an efficient and robust access method for points and rectangles. SIGMOD Rec. 19 (2), 322–331.

Blue, V. J., Adler, J. L., 2001. Cellular automata microsimulation for modeling bi-directional pedestrian walkways. Transportation Research Part B: Methodological 35 (3), 293–312.

Burstedde, C., Klauck, K., Schadschneider, A., Zittartz, J., 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. Physica A: Statistical Mechanics and its Applications 295 (3-4), 507–525.

Chraibi, M., Kemloh, U., Schadschneider, A., Seyfried, A., 2011a. Force-based models of pedestrian dynamics. Netw. Heterog. Media 6, 425–442.

Chraibi, M., Seyfried, A., Schadschneider, A., 2010. Generalized centrifugal force model for pedestrian dynamics. Phys. Rev. E 82, 046111.

Chraibi, M., Wagoum, U. K., Schadschneider, A., Seyfried, A., 2011b. Force-based models of pedestrian dynamics. Networks and Heterogeneous Media 6, 425–442.

Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1 (1), 269.

Dijkstra, J., Jessurun, A. J., Vries, B. d., Timmermans, H. J. P., 2006. Agent Architecture for Simulating Pedestrians in the Built Environment. In: Bazzan, A. L., Chaib-draa, B., Klügl, F., Ossowski, S. (Eds.), Forth Internation Workshop on Agents in Traffic and Transportation (ATT2006). New York, pp. 8–16.

Ding, Q., Wang, X., Shan, Q., Zhang, X., 2011. Modeling and Simulation of Rail Transit Pedestrian Flow. Journal of Transportation Systems Engineering and Information Technology 11 (5), 99–106.

Funge, J., Tu, X., Terzopoulos, D., 1999. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In: Proceedings of the 26th annual conference on computer graphics and interactive techniques. SIGGRAPH '99. ACM Press/Addison-Wesley Publishing Co, New York and NY and USA, pp. 29–38.

Gloor, C., Stucki, P., Nagel, K., 2004. Hybrid Techniques for Pedestrian Simulations. In: Sloot, P., Chopard, B., Hoekstra, A. (Eds.), Cellular Automata. Vol. 3305 of Lecture Notes in Computer Science. Springer, pp. 581–590.

Golledge, R. G., 1999. Wayfinding behavior: Cognitive mapping and other spatial processes. Johns Hopkins Univ. Press, Baltimore.

Gunnar G., L., 1998. Models of wayfinding in emergency evacuations. European Journal of Operational Research 105 (3), 371–389.

Guo, R.-Y., Huang, H.-J., 2012. Formulation of pedestrian movement in microscopic models with continuous space representation. Transportation Research Part C: Emerging Technologies 24 (0), 50–61.

Guo, R.-Y., Huang, H.-J., Wong, S. C., 2011. Collection, spillback, and dissipation in pedestrian evacuation: A network-based method. Transportation Research Part B: Methodological 45 (3), 490–506.

Haklay, M., O'Sullivan, D., Thurstain-Goodwin, M., Schelhorn, T., 2001. "So go downtown": simulating pedestrian movement in town centres. Environment and Planning B: Planning and Design 28 (3), 343–359.

Hamacher, H. W., Tjandra, S. A., 2002. Mathematical modelling of evacuation problems: A state of the art. In: Schreckenberg, M. (Ed.), Pedestrian and Evacuation Dynamics. Springer, Berlin, pp. 227–266.

Hartmann, D., 2010. Adaptive pedestrian dynamics based on geodesics. New Journal of Physics 12 (4), 043032.

Helbing, D., 1992. A fluid-dynamic model for the movement of pedestrians. Complex Systems 6, 391–415.

Helbing, D., Molnár, P., 1995. Social Force Model for Pedestrian Dynamics. Physical Review E 51 (5), 4282–4286.

Henderson, L. F., 1974. On the fluid mechanics of human crowd motion. Transportation Research Part B: Methodological 8 (6), 509–515.
Höcker, M., Berkhahn, V., Kneidl, A., Borrmann, A., Klein, W., 2010. Graph-based approaches for simulating pedestrian dynamics in building models. In: University College Cork (Ed.), 8th European Conference on Product & Process Modelling (ECPPM). Cork and Ireland.
Hoogendoorn, S. P., Bovy, P. H. L., 2004. Pedestrian route-choice and activity scheduling theory and models. Transportation Research Part B: Methodological 38 (2), 169–190.
Keil, J. M., Gutwin, C. A., 1992. Classes of Graphs Which Approximate the Complete Euclidean Graph. Discrete & Computational Geometry 7, 13–28.
Kirik, E., Yurgel'yan, T. B., Krouglov, D., 2009. The shortest time and/or the shortest path strategies in a CA FF pedestrian dynamics model. J. Siberian Fed. Univ. Math. Phys. 2 (3), 271–278.
Klüpfel, H., 2003. A Cellular Automaton Model for Crowd Movement and Egress Simulation. Ph.D. thesis, Universität Duisburg-Essen, Duisburg.
Klüpfel, H., Meyer-König, T., 2004. Simulation of the evacuation of a football stadium. In: S. Hoogendoorn et. al (Ed.), Traffic and Granular Flow '03. Springer, Berlin, pp. 423–430.
Kneidl, A., Borrmann, A., 2011. How Do Pedestrians find their Way? Results of an experimental study with students compared to simulation results. In: Jaskolowski, W., Kepka, P. (Eds.), EMEVAC. The Main School of Fire Service, Warsaw.
Kneidl, A., Borrmann, A., Hartmann, D., 2011. Generating sparse navigation graphs for microscopic pedestrian simulation models. In: Hartmann, T., de Wilde, P., Rafiq, Y. (Eds.), Proceedings of the 2011 eg-ice Workshop.
Köster, G., Hartmann, D., Klein, W., 2010. Microscopic pedestrian simulations: From passenger exchange times to regional evacuation. In: Operations Research - Mastering complexity.
Kretz, T., 2009. Pedestrian traffic: on the quickest path. Journal of Statistical Mechanics: Theory and Experiment 2009 (03), P03012.
Kretz, T., Hengst, S., Rocea, V., Perez Arias, A., Friedberger, S., Hanebeck, U. D., 2011. Calibrating Dynamic Pedestrian Route Choice with an Extended Range Telepresence System. In: 1st IEEE Workshop on Modeling, Simulation and Visual Analysis of Large Crowds.
Kretz, T., Schreckenberg, M., 2006. F.A.S.T. - Floor field- and Agent-based Simulation Tool. In: Chung, E., Barcel'o, J., Dumont, A.-G. (Eds.), International Symposium of Transport Simulation (ISTS06).
Lakoba, T., Kaup, D., Finkelstein, N., 2005. Modifications of the Helbing-Molnár-Farkas-Vicsek social force model for pedestrian evolution. Simulation 81, 339–352.
Lämmel, G., Rieser M., Nagel, K., 2010. Large Scale Microscopic Evacuation Simulation. In: Klingsch, W. (Ed.), Pedestrian and evacuation dynamics 2008. Springer, Berlin and Heidelberg, pp. 547–553.
Lerner, A., Chrysanthou, Y., Lischinski, D., 2007. Crowds by Example. Computer Graphics Forum 26 (3), 655–664.
Löhner, R., 2010. On the modeling of pedestrian motion. Appl. Math. Model. 34, 366–382.
Loscos, C., Marchal, D., Meyer, A., 2003. Intuitive Crowd Behaviour in Dense Urban Environments using Local Laws. In: University of Birmingham (Ed.), Theory and Practice of Computer Graphics, 2003. Proceedings,. IEEE Computer Society, Los Alamitos and Calif., pp. 122–129.
Mayer, H., Hartmann, D., Klein, W., 2011. Coupling pedestrian simulation and fire simulation for online evacuation planning. In: Jaskolowski, W., Kepka, P. (Eds.), EMEVAC. The Main School of Fire Service, Warsaw.
Mitchell, D. H., MacGregor Smith, J., 2001. Topological network design of pedestrian networks. Transportation Research Part B: Methodological 35 (2), 107–135.
Nishinari, K., Kirchner, A., Namazi, A., Schadschneider, A., 2004. Extended floor field CA model for evacuation dynamics. IEICE Trans. Inf. Syst E87D, 726–732.
Parisi, D. R., Gilman, M., Moldovan, H., 2009. A modification of the social force model can reproduce experimental data of pedestrian flows in normal conditions. Physica A 388, 3600–3608.
Pelechano, N., Malkawi, A., 2008. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. Automation in Construction 17 (4), 377–385.
Petres, C., Pailhas, Y., Patron, P., Petillot, Y., Evans, J., Lane, D., 2007. Path Planning for Autonomous Underwater Vehicles. IEEE Transactions on Robotics 23, 331–341.
Reynolds, C., 1999. Steering Behaviors for Autonomous Characters. In: Game Developers Conference.
Rindsfüser, G., Klügl, F., 2007. Agent-Based Pedestrian Simulation: A Case Study of the Bern Railway Station. In: Nagel, K., Koll-Schretzenmayr, M. (Eds.), disP. Vol. 170. ETH Zürich, Zurich and Switzerland, pp. 9–18.
Robin, T., Antonini, G., Bierlaire, M., Cruz, J., 2009. Specification, estimation and validation of a pedestrian walking behavior model. Transportation Research Part B: Methodological 43 (1), 36–56.
Rodriguez, S., Amato, N. M., 2010. Behavior-Based Evacuation Planning. In: Proceedings of 2010 IEEE International Conference on Robotics and Automation. Anchorage.
Ronald, N., Sterling, L., Kirley, M., 2007. An Agent-Based Approach To Modelling Pedestrian Behaviour. Lecture Notes in Computer Science, 145–156.

Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., Seyfried, A., 2009. Evacuation dynamics: Empirical results, modeling and applications. Encyclopedia of Complexity and System Science, 3142–3176.

Sethian, J. A., 1999. Level Set Methods and Fast Marching Methods. Cambridge University Press.

Shi, C., Zhong, M., Nong, X., He, L., Shi, J., Feng, G., 2012. Modeling and safety strategy of passenger evacuation in a metro station in China. Safety Science 50 (5), 1319–1332.

Sud, A., Andersen, E., Curtis, S., Lin, M. C., Manocha, D., 2008. Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs. IEEE Transactions on Visualization and Computer Graphics 14, 526–538.

Teknomo, K., 2008. Modeling mobile traffic agents on network simulation. In: Transportation Science Society of the Phillippines 2008 – Proceedings of the 16th Annual of Transportation Science.

Teknomo, K., Fernandez, P., 2012. Simulating optimum egress time. Safety Science 50 (5), 1228–1236.

Teknomo, K., Millonig, R., 2007. A navigation algorithm for pedestrian simulation in dynamic environments. In: Proceedings of the 11th World Conference on Transport Research (WCTR). Berkeley and USA.

Varas, A., Cornejo, M. D., Mainemer, D., Toledo, B., Rogan, J., Munoz, V., Valdivia, J. A., 2007. Cellular automaton model for evacuation process with obstacles. Physica A: Statistical Mechanics and its Applications 382 (2), 631–642.

Wagoum, U. K., Seyfried, A., Holl, S., 2012. Modelling dynamic route choice of pedestrians to assess the criticality of building evacuation. Advances in Complex Systems.

Weidmann, U., 1993. Transporttechnik der Fussgänger: Transporttechnische Eigenschaften des Fussgängerverkehrs (Literaturauswertung).

Yamamoto, K., Kokubo, S., Nishinari, K., 2007. Simulation for pedestrian dynamics by real-coded cellular automata. Physica A: Statistical Mechanics and its Applications 379 (2), 654.

Yu, W., Chen, R., Dong, L., Dai, S., 2005. Centrifugal force model for pedestrian dynamics. Phys. Rev. E 72, 026112.

Zhang, Q., Han, B., Li, D., 2008. Modeling and simulation of passenger alighting and boarding movement in Beijing metro stations. Transportation Research Part C: Emerging Technologies 16 (5), 635–649.